Halaman Judul

LAPORAN PENELITIAN

IMPLEMENTASI TEKNOLOGI FAIL2BAN UNTUK PERLINDUNGAN SERVER MAIL

oleh
W A G I T O, S.T., M.T.
NIDN : 0522126901
NPP : 961080

Mendapat Bantuan Biaya Penelitian dari Puslitbang dan PPM
Semester Genap 2017/2018

Sekolah Tinggi Manajemen Informatika dan Komputer
AKAKOM YOGYAKARTA
Tahun 2018

HALAMAN PENGESAHAN

1.	a. Judul Penelitian	:	Implementasi Teknologi Fail2ban Untuk Perlindungan *server mail*
	b. Bidang Ilmu	:	Jaringan Komputer
	c. Kategori	:	Implementasi Jaringan Komputer
2.	Ketua Peneliti		
	a. Nama	:	Wagito, S.T., M.T.
	b. NIDN	:	0522126901
	c. NPP	:	961080
	d. Pangkat/Golongan	:	Pembina Tk 1 / IV B
	e. Jabatan Fungsional	:	Lektor Kepala
	f. Jurusan/Prodi	:	Teknik Informatika
	g. Alamat Institusi	:	Jalan Raya Janti Karang Jambe, Yogyakarta
5.	Waktu Penelitian	:	6 bulan
6.	Biaya Penelitian	:	

Yogyakarta, Oktober 2018
Mengetahui

Ketua Prodi

Dini Fakta Sari S.T., M.T.
121172/0507108401

Ketua Peneliti

Wagito, S.T., M.T.
961080/0522126901

Menyetujui
Kepala Puslit dan PPM
STMIK AKAKOM

Edy Prayitno, S.Kom., M.Eng
151185/0502117203

Kata Pengantar

Puji syukur kepada Allah S.W.T. karena telah melimpahkan rahmat, hidayah dan taufik-Nya Berkat pertolongan dan tuntunan-Nya serta dengan berbagai usaha akhirnya penelitian ini berhasil diselesaikan dengan baik.

Penelitian yang berjudul implementasi teknologi Fail2ban untuk perlindungan *server mail* untuk meneliti cara melindungi *server mail* dari gangguan. Gangguan *server mail* antara lain penggunaan username dan password untuk menyebarkan *email* spam ke Internet. Jika *server mail* dianggap sebagai penyebar *email* spam, maka *server mail* akan dikenakan blok oleh *server mail* lain di Internet. Hal demikian menjadikan reputasi domain menjadi tidak baik.

Hasil penelitian ini masih banyak kekurangannya, sehingga kritik dan saran yang membangun untuk lebih mengembangkan hasilnya sangat diharapkan. Semoga hasil penelitian ini bermanfaat bagi semua orang.

Penulis

Daftar Isi

Daftar Gambar

ABSTRAK

Email menjadi sarana terbesar kedua setelah *web* yang digunakan oleh pengguna dalam berhubungan dengan dunia maya Internet. Dalam perkembangannya *email* dapat digunakan untuk mengirim banyak jenis *file*. Untuk dapat memberi layanan *email* diperlukan *server mail*. *Server mail* merupakan suatu sistem *server* yang melibatkan beberapa fungsi yaitu fungsi transfer *email* dan fungsi penyimpanan *email*. Dalam operasi, *server mail* banyak mengalami gangguan yang berasal dari pengirim *spam*.

Penelitian mencoba untuk membuat suatu metode untuk mengamankan *server mail* dari usaha untuk menerobos *password*. Pengamanan ini merupakan tingkat usaha pertama untuk mengatasi upaya *brute force password* yang dilakukan pengguna ilegal. Penelitian dilakukan dengan cara membuat dua *server mail*. Satu *server* sebagai uji konfigurasi, *server* lain dipakai sebagai penerima *email*. Perangkat lunak yang digunakan adalah Postfix sebagai MTA, Dovecot sebagai penyedia layanan IMAP dan POP3 serta Fail2ban sebagai pengendali lalu-lintas *email* pada *server mail*. Uji *server mail* dilakukan untuk gangguan keamanan *password* dan gangguan penyebaran *spam*.

Penelitian berhasil menyusun metode untuk melindungi *server mail* dari gangguan, menerapkan perangkat lunak Fail2ban untuk melindungi *server mail*, menerapkan metode pengamanan *password* pada operasi *server mail* sesungguhnya yang mana. Untuk mengatur kecepatan pengiriman *email* keluar disarankan pengaturan langsung pada Postfix, tidak menggunakan Policyd. Penelitian belum sampai menyusun metode deteksi untuk melihat apakah keamanan *server mail* berhasil diterobos.

Kata kunci: pengendalian, keamanan, *server*, *mail*, Fail2ban

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Surat merupakan sarana komunikasi antar manusia yang sudah lama digunakan. Komunikasi dilakukan dalam bentuk pesan tulisan dari pengirim ke penerima. Untuk mengirim surat dibutuhkan alamat penerima serta biaya yang diwujudkan dalam bentuk prangko. Umumnya pesan dituliskan dalam kertas yang dimasukkan dalam amplop kemudian dititipkan pada penyedia layanan surat. Penyedia layanan surat di Indonesia dilakukan oleh kantor pos.

Sejalan dengan perkembangan jaringan komputer maka berkembang juga jaringan Internet. Manusia dapat berkomunikasi menggunakan beberapa layanan yang disediakan pada Internet. Salah satu layanan yang disediakan oleh jaringan Internet adalah surat elektronik (*electronic mail*) atau yang dikenal dengan nama *email*. *Email* pada dasarnya merupakan pengganti pengiriman pesan melewati surat kertas.

Email menjadi sarana terbesar kedua setelah *web* yang digunakan oleh pengguna dalam berhubungan dengan dunia maya Internet. Pada mulanya *email* hanya berfungsi menggantikan pesan teks seperti pada pengiriman surat lewat pos.

Dalam perkembangannya *email* tidak hanya dapat digunakan mengirim pesan teks, namun dapat digunakan untuk mengirim dokumen elektronik secara umum baik teks, gambar, suara, dan sebagainya. Dengan demikian, pengguna semakin dimudahkan dalam menggunakan layanan *email* untuk mengirimkan dokumen elektronik.

Untuk dapat berkomunikasi menggunakan *email* diperlukan perangkat lunak klien *mail* dan *server mail*. Klien *mail* adalah program yang digunakan untuk mengakses *email* baik membaca maupun mengirim *email*. Beberapa perangkat lunak yang berfungsi sebagai klien *mail* antara lain Ms. Outlook, KMail, Thunderbird dan sebagainya. Klien *mail* membaca *email* menggunakan protokol baik IMAP atau POP. Untuk mengirimkan *email*, digunakan protokol SMTP. Variasi protokol yang dipakai adalah IMAPS, POPS dan SMTPS yang melibatkan fitur keamanan data pada saat mengakses *email*.

Server mail merupakan suatu sistem *server* yang melibatkan beberapa fungsi yaitu fungsi transfer *email* dan fungsi penyimpanan *email*. Salah satu perangkat lunak yang dapat digunakan sebagai agen transfer *email* adalah Postfix. Postfix berlaku sebagai *server* SMTP. Salah satu perangkat lunak yang dapat digunakan untuk menyimpan *email* adalah Dovecot. Dovecot menyediakan beberapa protokol untuk akses *email* antara lain IMAP dan POP.

Dalam operasi, *server mail* banyak mengalami gangguan yang berasal dari pengirim *spam* atau yang disebut *spammer*. Gangguan *spammer* dapat dilihat dari dua sisi. Pada sisi pertama *spammer* mengirimkan *email spam* menuju *server mail*

(*inbound*) yang menyebabkan pengguna *server mail* menerima *email spam*. Pada sisi yang kedua, *spammer* memanfaatkan *server mail* untuk mengirim *email spam* ke alamat *email* di luar (*outbound*).

Pada sisi pengguna layanan *email*, *spammer* dikatakan mengganggu, karena mengirim *email spam* dalam jumlah banyak ke alamat *email* secara acak. *Email spam* menyebabkan *inbox* pengguna menjadi penuh dengan *email spam*. Akses *email* pengguna menjadi sangat lambat akibat *inbox* yang penuh dengan *email spam*. Apabila jumlah *email* sangat banyak, proses penghapusannya juga sangat merepotkan.

Pada sisi pengelolaan *server mail*, *spammer* menyebabkan kerja *server* menjadi lebih berat yang pada akhirnya akan mengganggu *email* legal yang menjadi tanggung jawab *server mail*. Karena alamat *email* yang dipakai *spammer* acak, banyak alamat *email* yang dipakai tidak legal. Hal demikian menyebabkan munculnya antrean pengiriman *email* pada *server*. Antrean ini harus selalu dibersihkan secara periodik, karena tidak akan bisa terkirim oleh *server mail*. Yang sangat merugikan adalah bahwa domain *server mail* dianggap sebagai sumber *spam*. Hal ini menjadikan alasan pihak lain untuk melakukan blok *email* dari *server mail*.

penggunaan *server mail* oleh *spammer* disebabkan oleh sistem keamanan yang diterapkan masih lemah. Sumber kelemahan bisa berasal dari dalam sistem *server mail*, yang mana *server* mengizinkan akses langsung pada *server* tanpa melalui *password*. Untuk mengatasi sumber kelemahan ini dengan menerapkan

password ketika akses pada *server mail*. Penerapan *password* pun juga masih punya kelemahan apabila pengguna *email* menggunakan *password* yang lemah. *Password* bisa diterapkan pada saat masuk sistem untuk akses *email* masuk dan pada saat pengiriman *email*. Apabila *server mail* menerapkan *password* untuk akses *email*, salah satu jalan pengguna ilegal *server mail* adalah dengan menerobos keamanan *password*.

Untuk menerobos keamanan *password server mail*, pengguna ilegal biasanya memanfaatkan kamus *password* yang biasanya beredar luas di Internet. Apabila *password* yang dipakai pengguna *email* cukup lemah, biasanya bisa diterobos menggunakan kamus *password* tersebut. Kamus *password* ini berisi daftar *password* yang sudah diatur sebelumnya dan bisa didasarkan pada kamus kata. Cara ini relatif terstruktur dalam usaha untuk menerobos keamanan *password* karena didasarkan pada daftar kata yang ada pada kamus *password*.

Apabila cara terstruktur menerobos *password* gagal dilakukan, pengguna ilegal menggunakan dengan metode *brute force password*. Metode *brute force password* adalah cara untuk menerobos *password* dengan mencoba seluruh kemungkinan *password* baik kombinasi huruf dan angka serta ukuran *password* sampai ditemukan *password* yang sesuai. Cara *brute force password* menjadi jalan terakhir apabila cara terstruktur untuk menerobos *password* tidak dapat dilakukan.

Penelitian mencoba untuk membuat suatu metode untuk mengamankan *server mail* dari usaha untuk menerobos *password*. Pengamanan ini merupakan tingkat usaha pertama untuk mengatasi upaya *brute force password* yang

dilakukan pengguna ilegal. Apabila keamanan *password* berhasil ditembus, maka perlu dilakukan langkah untuk mengamankan *server mail*. Untuk melakukan pengamanan lanjutan perlu diteliti perilaku yang dilakukan pengguna ilegal setelah berhasil akses. Salah satu yang bisa dilakukan pengguna ilegal setelah berhasil memperoleh akses adalah menggunakan *server mail* sebagai sumber *spam*. Dalam hal ini perlu dilakukan deteksi terhadap penggunaan ilegal pada *server mail*.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian adalah bagaimana menerapkan perangkat lunak Fail2ban untuk melindungi *server email* sedemikian, sehingga dapat mencegah dan mengatasi gangguan keamanan yang berkaitan dengan *email*.

1.3 Batasan Masalah

Pada penelitian ini ditetapkan batasan masalah berkaitan dengan cukup luasnya lingkup yang bisa dicakup penelitian. Batasan masalah yang ditetapkan pada penelitian ini adalah sebagai berikut.

1. Pengamatan gangguan keamanan *email* didasarkan pada catatan *log* yang dihasilkan *server email*.
2. Pencegahan gangguan keamanan dilakukan pada protokol operasi *email* IMAP, POP dan SMTP.

3. Perangkat lunak *server email* yang berlaku sebagai agen transfer *email* yang digunakan untuk percobaan adalah Postfix.
4. Perangkat lunak yang berlaku sebagai *server* IMAP dan POP adalah Dovecot.
5. Perangkat lunak untuk menentukan apakah ada usaha gangguan yang digunakan adalah Fail2ban.
6. Pemblokiran dilakukan terhadap asal alamat IP yang dipakai pihak pengganggu keamanan.
7. Pemblokiran alamat IP pengganggu dilakukan dengan bantuan perangkat lunak Firewall Iptables.
8. Pengamatan terhadap lalu-lintas *email* dilakukan dengan bantuan perangkat lunak Policyd.
9. Perangkat lunak Policyd digunakan juga untuk mengatasi gangguan apabila berhasil menembus keamanan Fail2ban.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai pada penelitian implementasi teknologi Fail2ban untuk perlindungan *server mail* adalah sebagai berikut.

1. Menyusun metode untuk melindungi *server mail* dari gangguan yang berkaitan dengan *email*.
2. Menerapkan perangkat lunak Fail2ban untuk melindungi *server mail* dari

gangguan keamanan terutama gangguan *brute force password*.

3. Menyusun metode deteksi untuk melihat apakah keamanan *server mail* berhasil diterobos.
4. Menyusun metode untuk mengatasi pengguna ilegal yang berhasil menerobos keamanan *password*.
5. Menerapkan metode pengamanan *password* pada operasi *server mail* sesungguhnya.

1.5 Manfaat Penelitian

Penelitian yang dihasilkan bermanfaat bagi pengelola *server* untuk menentukan metode perlindungan terhadap *server mail* yang menjadi tanggung jawabnya. Bagi para pemakai layanan *email*, dapat memperoleh manfaat dari layanan *email* yang aman.

1.6 Target Luaran

Hasil penelitian direncanakan dilakukan publikasi dan seminar pada kegiatan ilmiah. Publikasi yang diinginkan adalah pada jurnal ilmiah nasional yang sudah akreditasi. Jika sulit masuk pada jurnal akreditasi, maka paling tidak bisa masuk jurnal ilmiah nasional.

BAB 2 TINJAUAN PUSTAKA

Beberapa penelitian berkaitan dengan kecepatan *transfer* data telah dilakukan dan dipublikasikan dalam bentuk jurnal. Penelitian-penelitian tersebut berbeda-beda pada objek serta topik yang diteliti.

Hasil penelitian tentang keamanan *server mail* pernah dipublikasikan melalui *e-Proceeding of Applied Science* (Galih Dwiyan Prakoso, dkk., 2017). Penelitian ini menggunakan *server mail* Zimbra. *Server mail* dikombinasikan dengan menerapkan *spamassassin*, *whitelist* dan *blacklist*. Pengamanan spam dilakukan pada log yang dihasilkan *server*. *Spamassassin* dan *whitelist* pada sistem *server mail* Zimbra digunakan untuk mencegah *spam* dengan cara penerapan *spamassassin list*, *whitelist user* dan *blacklist user* yang dilakukan konfigurasi pada Zimbra.

Hasil penelitian berkaitan dengan keamanan *server mail* dipublikasikan pada Seminar Ilmiah Sistem Informasi dan Teknologi Informasi (Dandy Pramana Hostiadi, 2016). Penelitian menggunakan perangkat lunak Zimbra sebagai *server mail*. Penelitian dilakukan menggunakan metode *pretty good privacy* (PGP). Hasil penelitian menunjukkan bahwa PGP mampu mengamankan *email* baik teks dan *attachment*, menunjukkan perbedaan ukuran *file attachment* lebih besar dan

mengubah *header mail* dari *mail* standar.

Tulisan tentang penggunakan Fail2ban untuk melindungi SSH *server* dipublikasikan di Internet (Supriyo Biswas, 2018). Fail2ban melindungi *server* SSH dengan segera. Dengan sedikit konfigurasi, itu bisa membuat serangan brute force yang masif menjadi masalah yang sepele.

Tulisan berikutnya yang menyajikan bagaimana melindungi *server* SSH dari serangan *brute force* pernah dipublikasikan melalui Internet (Dan Nanni, 2013). Pada tulisan ini ditunjukkan cara instalasi dan konfigurasi Fail2ban untuk melindungi *server* SSH. Meskipun Fail2ban dapat mengurangi serangan menebak kata sandi brute force, perlu dicatat bahwa ini tidak dapat melindungi *server* SSH terhadap kampanye *brute force* terdistribusi canggih, di mana penyerang melewati Fail2ban dengan menggunakan ribuan alamat IP yang dikontrol melalui robot (*bot-controlled IP addresses*).

Tulisan lain juga menyajikan bagaimana Fail2ban digunakan untuk melindungi *server* SSH dari serangan *brute force* (Matthew Setter, 2017). Fail2ban - terutama dengan konfigurasi minimal ini - tidak memberikan perlindungan seratus persen terhadap serangan di situs *web* dan *server*. Namun, ini cara terbaik untuk memulai. Jika tidak ada yang lain, dengan Fail2ban dimiliki perlindungan yang jauh lebih besar daripada perlindungan lain, dan pengelola *server* dibuat sadar tentang berapa banyak serangan yang dilakukan terhadap *server* setiap jam demi jam.

BAB 3 TEORI

3.1 Postfix

Postfix ditulis oleh Wietse Venema, yang dikenal luas untuk *tool* keamanan dan dokumen. Postfix dibuat tersedia sebagai perangkat lunak open *source* pada bulan Desember 1998. IBM *Research* mensponsori rilis awal dan terus mendukung pengembangan yang sedang berlangsung. (IBM menyebut paket *Secure Mailer*.) Ada beberapa tujuan dari awal yang mendorong desain dan pengembangan Postfix: (Kyle D , 2003)

1. Keandalan: Postfix menunjukkan nilai sebenarnya ketika beroperasi di bawah kondisi yang penuh tekanan. Meskipun dalam lingkungan yang sederhana, perangkat lunak dapat mengalami kondisi yang tidak terduga.
2. Keamanan: Postfix mengasumsikan itu berjalan di lingkungan yang tidak bersahabat. Postfix menggunakan berbagai lapisan pertahanan untuk melindungi terhadap penyerang. Konsep keamanan dari hak akses yang paling rendah digunakan pada seluruh sistem Postfix, sehingga setiap proses, yang dapat dijalankan dalam kompartemen yang terisolasi, berjalan dengan seperangkat hak akses terendah yang dibutuhkannya.

3. Kinerja: Postfix ditulis dengan kinerja dalam pikiran dan, pada kenyataannya, mengambil langkah-langkah untuk memastikan bahwa kecepatannya tidak membanjiri sistem lain. Ini menggunakan teknik untuk membatasi baik jumlah proses baru yang harus dibuat dan jumlah akses sistem *file* yang dibutuhkan dalam memproses pesan.
4. Fleksibilitas: Sistem Postfix sebenarnya terdiri dari beberapa program dan sub sistem yang berbeda. Pendekatan ini memungkinkan fleksibilitas yang tinggi. Semua bagian mudah dipantau melalui *file* konfigurasi secara langsung.
5. Kemudahan penggunaan: Postfix adalah salah satu paket *email* yang lebih mudah untuk diatur dan dikelola, karena menggunakan *file* konfigurasi langsung dan tabel *lookup* sederhana untuk translasi dan penerusan alamat.
6. Kompatibilitas dengan Sendmail: Dengan kompatibilitas Sendmail, Postfix dapat dengan mudah mengganti Sendmail pada sistem tanpa memaksa perubahan apa pun pada pengguna atau merusak aplikasi yang bergantung padanya.

Postfix mempunyai beberapa fitur utama yaitu: dukungan kontainer, kontrol *junk mail*, dukungan protokol, dukungan *mailbox*, dukungan basis data, dan manipulasi alamat. Beberapa fitur memerlukan pustaka pihak ketiga (contoh: LDAP, SQL, TLS). Fitur lain hanya tersedia ketika dukungan sistem operasi yang diperlukan ada dan Postfix mengetahui cara menggunakannya (contoh: IP versi 6, koneksi

caching) (Postfix, 2018).

3.2 Dovecot

Dovecot adalah *open source* IMAP dan *server* POP3 untuk sistem Linux/UNIX-like, ditulis dengan keamanan terutama dalam pikiran. Dovecot adalah pilihan yang sangat baik untuk instalasi kecil dan besar. Ini cepat, mudah diatur, tidak memerlukan administrasi khusus dan hanya menggunakan sedikit memori. (Timo Sirainen, 2017)

Ada beberapa opsi yang dapat digunakan untuk mengatur bagaimana Dovecot dijalankan. (Timo Sirainen, 2017)

1. -a: *Dump* semua pengaturan konfigurasi untuk *stdout* dan keluar dengan sukses. Sama seperti doveconf -a.
2. -c config-file: Mulai Dovecot dengan konfigurasi alternatif.
3. -F: Jalankan Dovecot di *foreground*, jangan *daemonize*.
4. -n:Dump pengaturan *non-default* ke *stdout* dan keluar dengan sukses. Sama seperti doveconf-n.
5. -p: Prompt untuk kata sandi kunci ssl untuk ssl_key yang dilakukan konfigurasi saat *startup*.
6. --build-options: Tampilkan opsi *build* Dovecot dan keluar dengan sukses.
7. --help: Cetak pesan penggunaan untuk *stdout* dan keluar dengan sukses.
8. --hostdomain: Menunjukkan nama *host.domain* saat ini dari sistem. Jika

pencarian domain gagal karena beberapa alasan, hanya nama *host* yang akan ditampilkan.

9. --version: Tampilkan versi Dovecot dan keluar dengan sukses.

Ada dua variasi perintah yang dapat digunakan untuk menjalankan Dovecot sesuai kondisi terakhir.

1. reload: Paksa Dovecot memuat ulang konfigurasi.
2. stop: *Shutdown* Dovecot dan semua proses anaknya.

Ketika shutdown_clients diatur ke no, sesi yang sudah ada akan terus menggunakan pengaturan lama, setelah *reload* Dovecot. Juga semua sesi akan tetap hidup setelah berhenti Dovecot. Secara bawaan semua sesi aktif akan dimatikan.

Dovecot dapat menangani beberapa sinyal khusus ketika sedang dijalankan

1. HUP: Paksa Dovecot memuat ulang konfigurasi.
2. INT: *Shutdown* Dovecot dan semua proses anaknya.
3. TERM: *Shutdown* Dovecot dan semua proses anaknya.
4. USR1: Paksa Dovecot untuk membuka kembali semua *file log* yang dibuat konfigurasi (log_path, info_log_path, dan debug_log_path).

Selain itu Dovecot juga dapat menangani dua isyarat lain yaitu ALARM dan PIPE. Namun kedua isyarat tersebut diabaikan.

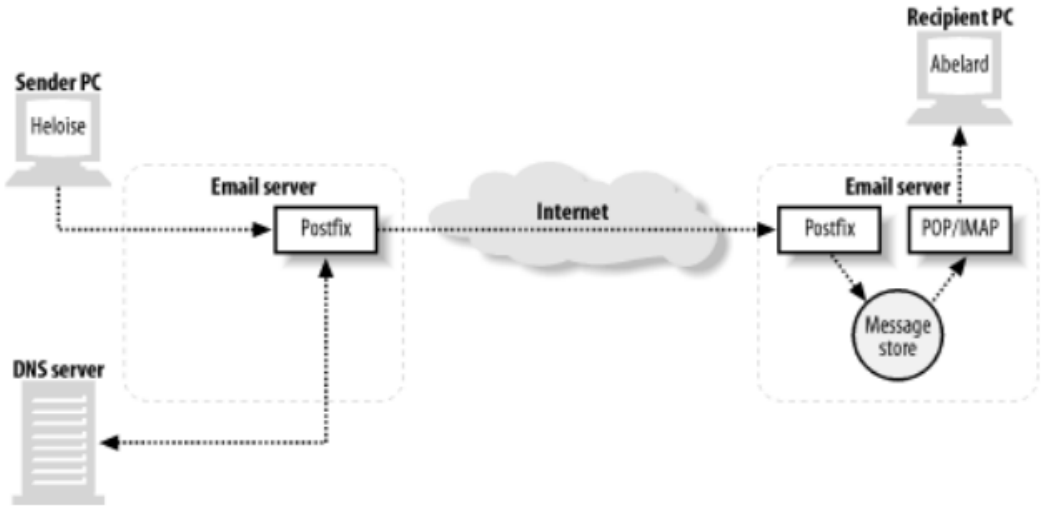
3.3 Server Mail

Server email (kadang-kadang juga disebut *server email*) adalah *server* yang menangani dan mengirim *email* melalui jaringan, biasanya melalui Internet. *Server email* dapat menerima *email* dari komputer klien dan mengirimnya ke *server email* lain. *Server email* juga dapat mengirim *email* ke komputer klien. (Mika Larramo, 2018). Supaya dapat berlaku sebagai *server mail*, *server* harus dipasang perangkat lunak yang dapat untuk menangani pengiriman dan penerimaan *email* seperti Postfix. Postfix berlaku sebagai MTA (*Mail Transfer Agent*). Postfix menggunakan protokol yang disebut SMTP (*Simple Mail Transfer Protocol*). Selain itu diperlukan perangkat lunak yang berlaku sebagai kotak penyimpan *email* dan melayani pembacaan *email* seperti Dovecot. Dovecot menggunakan protokol yang disebut POP (*Post Office Protocol*) atau IMAP (*Internet Message Access Protocol*). Ilustrasi proses pengiriman *email* dan posisi protokol-protokol yang berperan pada langkah-langkah pengiriman *email* ditunjukkan dalam Gambar 3.1.

Gambar 3.1 Proses Pengiriman *Email*

Ketika ditekan tombol "Kirim" dalam program *email*. Klien *email* akan terjadi hubungan ke *server* pada jaringan Internet yang disebut *server* SMTP. SMTP merupakan protokol yang digunakan ketika *email* dikirim dari klien ke *server* dan dari *server* ke *server* lain. Ketika *email* diunduh menuju program *email*, program tersebut akan terjadi hubungan pada *server* di Internet yang dikenal sebagai *server* POP3. *Server* POP3 menggunakan protokol bernama POP3 untuk komunikasi yang dilakukan. Selain menggunakan POP3 dapat juga digunakan protokol IMAP untuk melakukan proses unduh terhadap *email* dari *server* IMAP.

Peran Postfix dan Dovecot pada proses pengiriman dan penerimaan *email* ditunjukkan pada Gambar 3.2. Gambar 3.2 memberi ilustrasi contoh sederhana transmisi pesan di mana Postfix menangani tanggung jawab MTA dan pengiriman lokal. Sebagai MTA, Postfix menerima dan mengirim pesan *email* melalui jaringan melalui protokol SMTP. Untuk pengiriman lokal, agen pengiriman lokal Postfix dapat menyimpan pesan langsung kepada simpanan pesan atau memberikan pesan kepada agen pengiriman pesan khusus.



Gambar 3.2 Peran Postfix dan Dovecot

Postfix sebagai *server* SMTP pada kedua ujung transaksi *email.* Namun, karena Postfix didasarkan pada standar Internet, *server email* lain dapat dengan mudah menjadi *server* standar yang sesuai. Postfix dapat berkomunikasi dengan *server* lain yang menggunakan protokol SMTP (Kyle D , 2003). Ketika penerima akan membaca *email* yang disimpan pada kotak pesan, digunakan protokol POP/IMAP. Perangkat lunak yang berperan dalam proses pembacaan *email* oleh penerima adalah Dovecot.

3.4 Fail2ban

Aplikasi Fail2ban memonitor *file log server* untuk upaya intrusi dan aktivitas mencurigakan lainnya. Setelah sejumlah kegagalan yang ditetapkan

sebelumnya dari host, Fail2ban memblokir alamat IP secara otomatis untuk durasi tertentu. (Chris C, 2018)

Fail2Ban mampu mengurangi tingkat upaya autentikasi yang salah namun tidak dapat menghilangkan risiko yang diberikan oleh autentikasi yang lemah. Konfigurasi layanan untuk menggunakan hanya dua faktor atau mekanisme autentikasi publik/privat jika diinginkan untuk melindungi layanan.(Chris C, 2016,)

Fail2ban memindai *file log* dan melarang IP yang menunjukkan tanda-tanda jahat (terlalu banyak kegagalan kata sandi), mencari eksploit, dan lain-lain. Umumnya Fail2Ban kemudian digunakan untuk memperbarui aturan Firewall untuk menolak alamat IP untuk jumlah waktu tertentu, meskipun tindakan lain yang sewenang-wenang (misalnya mengirim *email*) juga bisa dibuat konfigurasi. Keluar dari kotak Fail2Ban dilengkapi dengan filter untuk berbagai layanan. (Nethesis Srl, 2017)

Fail2ban membaca *file* konfigurasi .conf terlebih dahulu, kemudian .local *file* menimpa pengaturan apa pun. Karena ini, semua perubahan pada konfigurasi umumnya dilakukan dalam *file* .local, meninggalkan *file* .conf yang tidak tersentuh.(Elle Krout, 2017) Ada dua *file* konfigurasi yang penting yaitu Fail2ban dan jail baik punya ekstensi .conf maupun .local.

File Fail2ban.conf berisi profil konfigurasi bawaan. Pengaturan bawaan akan memberi Anda pengaturan kerja yang wajar. Jika Anda ingin melakukan perubahan apa pun, sebaiknya lakukan dalam *file* terpisah, fail2ban.local, yang

mengesampingkan fail2ban.conf.(Chris C, 2018)

Dari sini, dapat dilakukan edit definisi pada fail2ban.local agar sesuai dengan konfigurasi yang diinginkan. Nilai-nilai yang dapat diubah adalah:(Elle Krout, 2017)

1. loglevel: Tingkat detail yang disediakan oleh *log* Fail2ban dapat ditetapkan ke 1 (kesalahan), 2 (peringatan), 3 (info), atau 4 (debug).
2. logtarget: Mencatat tindakan ke *file* tertentu. Nilai bawaan dari /var/log/fail2ban.log menempatkan semua *login* ke *file* yang ditentukan. Bergantian, Anda dapat mengubah nilainya menjadi:

   STDOUT: menghasilkan data apa pun

   STDERR: tampilkan semua kesalahan

   SYSLOG: *log* berbasis pesan

   FILE: output ke *file*
3. socket: Lokasi *file* soket.
4. pidfile: Lokasi *file* PID.

File jail.conf akan mengaktifkan Fail2ban untuk SSH secara bawaan untuk Debian dan Ubuntu, tetapi tidak CentOS. Semua protokol dan konfigurasi lain (HTTP, FTP, dan lain-lain.) diberi komentar. Jika diinginkan untuk mengubahnya, maka perlu dibuat jail.local untuk mengedit bagian tertentu.(Elle Krout, 2017)

Bagian [DEFAULT] berisi opsi global yang mungkin perlu disunting sesuai pengaturan yang diinginkan:(Chris C, 2018)

1. ignoreip: Opsi ini memungkinkan untuk menentukan alamat IP atau nama *host* yang gagal2ban akan abaikan. Misalnya, bagian ini dapat ditambahkan alamat IP rumah atau kantor, sehingga Fail2ban tidak mencegah akses *server* dari IP tersebut. Untuk menentukan banyak alamat, harus dipisahkan dengan spasi. Sebagai contoh:

   ignoreip = 127.0.0.1/8 93.184.216.34
2. bantime: Opsi ini menentukan dalam detik berapa lama alamat IP atau *host* diblokir. Nilai standar adalah 600 detik (10 menit).
3. maxretry: Opsi ini mendefinisikan jumlah kegagalan yang diizinkan *host* sebelum dilarang.
4. findtime: Opsi ini digunakan bersama dengan opsi maxretry. Jika *host* melebihi pengaturan maxretry dalam periode waktu yang ditentukan oleh opsi findtime, itu dilarang untuk jangka waktu yang ditentukan oleh opsi bantime.

3.5 PolicyD

Policyd adalah *daemon* kebijakan *anti-spam* untuk Postfix (ditulis dalam C) yang melakukan proses acak berbasis Greylisting, Pengirim- (amplop, SASL atau host/ip) (pada pesan dan/atau volume per unit waktu yang ditentukan), Pembatasan tingkat penerima, *Spamtrap monitoring*/daftar hitam, HELO *auto blacklisting* dan pencegahan pengacak HELO.(Zimbra, Inc, 2016)

PolicyD v2 (*codecamed* "cluebringer") adalah *server* kebijakan *multi-platform* untuk MTA populer. Daemon kebijakan ini dirancang sebagian besar untuk lingkungan *hosting mail* skala besar. Tujuan utamanya adalah untuk menerapkan sebanyak mungkin pemberantasan *spam* dan fitur pemenuhan *email*, sementara pada saat yang sama menjaga portabilitas, stabilitas, dan kinerja yang diperlukan untuk misi *email* penting saat ini. Sebagian besar ide dan metode yang diterapkan dalam PolicyD v2 berasal dari PolicyD v1 serta keterlibatan penulis yang lama dalam industri *hosting mail* skala besar. (AllWorldIT, 2018)

PolicyD punya beberapa fitur yang memudahkan untuk pengaturan lalu-lintas *email* masuk ke *server mail* maupun *email* keluar dari *server mail*.

1. Plugin: Fitur diterapkan menggunakan kerangka plugin fleksibel. Di bawah ini adalah *plugin* resmi saat ini:

   Protokol Plugin: *Plugin* protokol memberikan dukungan protokol untuk MTA yang berbeda.

   Postfix: Dukungan untuk protokol pendelegasian SMTPD Postfix

   Bizanga: Dukungan untuk Bizanga menggunakan protokol HTTP
2. Modul: Modul menyediakan berbagai fitur dan penyempurnaan ke Policyd.
3. Kontrol akses: Daftar kontrol akses. Kebijakan yang cocok dapat memiliki putusan HOLD, REJECT, DISCARD, FILTER, REDIRECT dan berisi data putusan opsional.
4. Amavis: Dukungan untuk Amavisd-new.

5. CheckHelo: Mendukung berbagai pemeriksaan HELO / EHLO.
6. CheckSPF: Mendukung berbagai pemeriksaan SPF.
7. Greylisting: Dukungan *Greylisting*.
8. Kuota: Bergulir kuota jendela.
9. Akuntansi: Akuntansi berbasis penggunaan (Didukung di: r343, v2.1.x).

Persyaratan untuk PolicyD

1. MySQL, PostgreSQL atau SQLite
2. Net::Server> = 0,96
3. Net::CIDR
4. Config::IniFiles (debian: libconfig-inifiles-perl, rpm: perl-Config-IniFiles)
5. Cache::FastmMap (debian: libcache-fastmmap-perl, rpm: perl-Cache-FastMmap)
6. Mail::SPF (diperlukan untuk modul CheckSPF)

BAB 4 METODE PENELITIAN

4.1 Bahan Penelitian

Untuk penelitian digunakan bahan berupa *log* sistem *server*. *File log* yang dipakai untuk penelitian terutama adalah adalah *log email*. Log *email* mencakup *log* akses SMTP, IMAP dan POP. Catatan *log* aktivitas *email* secara lengkap (*email* keluar, masuk, *bouncing* dan sebagainya) dicatat dalam suatu *file* teks /var/log/maillog.

Catatan *log email* terutama dihasilkan oleh perangkat lunak Postfix dan Dovecot ketika ada akses baik pada saat pengguna masuk sistem, membaca *email*, maupun saat mengirimkan *email* keluar dari *server mail*. Catatan *log email* yang dihasilkan berupa *file* teks yang dapat dilihat dan diolah menggunakan pengolah *file* teks.

Lalu-lintas *email* dapat diamati menggunakan catatan yang dihasilkan oleh perangkat lunak Policyd. Policyd menyimpan catatan lalu-lintas *email* dalam bentuk data MySQL. Pengamatan data MySQL dilakukan menggunakan perangkat lunak manajemen sistem basis data Adminer. Adminer punya banyak fitur yang memudahkan pengguna dalam mengelola sistem basis data baik

membaca, menulis, membuat mutakhir maupun menghapus data.

4.2 Alat

Alat yang digunakan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan dalam penelitian meliputi perangkat lunak sistem operasi, perangkat lunak pendukung sistem jaringan serta perangkat lunak untuk mempermudah kerja penelitian. Daftar perangkat lunak yang digunakan dalam penelitian sebagai berikut.

1. Sistem Operasi Linux Centos 7
2. *Server* SMTP Postfix
3. *Server* IMAP Dovecot
4. Klien *email* Thunderbird
5. VirtualBox
6. Sekrip Bash
7. Policyd
8. Perl
9. PHP
10. PHPMailer
11. MySQL
12. Fail2ban
13. Iptables

14. Rsyslog
15. Adminer
16. Midnight commander
17. Nginx

Perangkat keras yang digunakan dalam penelitian meliputi *server* maupun peralatan pendukung jaringan lain yang punya spesifikasi seperti pada daftar berikut.

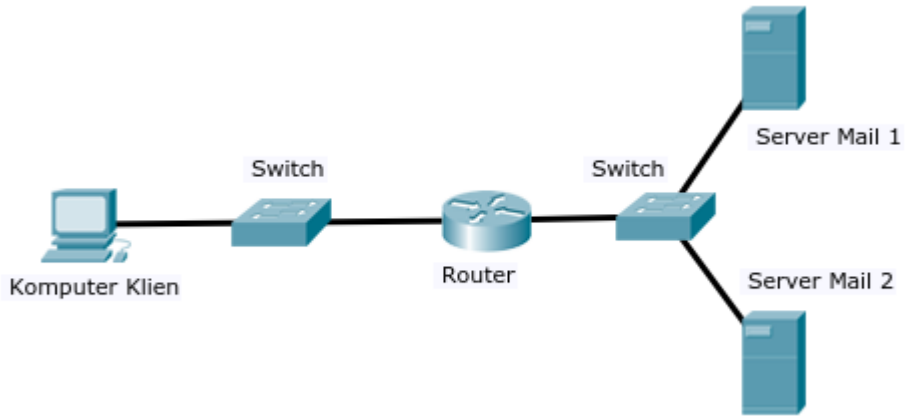
1. Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
2. RAM 4 GB.
3. Tipe sistem 64 bit.
4. Harddisk 500 GB.
5. Gigabit Ethernet
6. *Switch*
7. *Router*
8. Sistem Pengkabelan

4.3 Jalan Penelitian

4.3.1 Rancangan Perangkat Keras

Ada beberapa peralatan perangkat keras yang digunakan pada penelitian antara lain *server mail*, *switch*, *router* dan komputer klien. Peran *router* dalam

penelitian dapat digantikan fungsinya oleh komputer klien karena penelitian dilakukan menggunakan perangkat lunak VirtualBox. Diagram jaringan beserta peralatan-peralatan jaringan serta *server* yang digunakan dalam penelitian ditunjukkan pada Gambar 4.1.



Gambar 4.1 Jaringan Percobaan *Server Mail*

Fungsi masing-masing perangkat keras yang digunakan dalam penelitian dapat diuraikan sebagai berikut.

1. *Server mail* yang berfungsi sebagai *server* pengirim dan *server* penerima *email*. *Server mail* yang digunakan ada dua. Salah satu *server* digunakan untuk uji konfigurasi dan uji keamanan *server mail*, *Server mail* ini disebut saja *server mail* utama. *Server mail* yang kedua dipakai sebagai *server* pendamping *server mail* utama. Dua *server mail* ditempatkan pada jaringan yang berbeda supaya mirip dengan keadaan jaringan yang sesungguhnya.
2. *Router* difungsikan sebagai penghubung antara jaringan *server* dengan

jaringan komputer klien. *Router* akan meneruskan data dari jaringan *server* dan jaringan komputer klien.

3. *Switch* digunakan untuk menghubungkan komputer klien dengan *router* dan menghubungkan *router* dengan *server mail*. Peralatan *switch* bersifat pasif yang mana hanya digunakan menghubungkan beberapa peralatan pada sistem jaringan.
4. Komputer klien berfungsi sebagai komputer pengguna untuk menguji konfigurasi yang ada pada *server mail*. Komputer ini difungsikan layaknya komputer pengguna *email* legal dan pengguna *email* ilegal.

4.3.2 Rancangan Perangkat Lunak

Server mail menggunakan sistem operasi Linux Centos 7. Perangkat lunak MTA yang dipakai adalah Postfix. Postfix berfungsi sebagai *server* SMTP (*server mail*) yang bertugas untuk melayani transfer *email* dari dan menuju *server mail*. Selain itu, Postfix dapat juga difungsikan sebagai *relay* transfer *email*, namun pada penelitian ini, fungsi *relay* dibatasi untuk *email* yang berasal dari dalam *server mail*. Untuk *email* yang berasal dari luar *server mail*, dilakukan penutupan akses. Postfix merupakan perangkat lunak utama untuk proses pengiriman dan penerimaan *email*.

Selain perangkat lunak *server mail*, pada *server* juga dipasang perangkat lunak untuk pengamatan lalu-lintas *email* Policyd. Perangkat lunak ini

membutuhkan beberapa perangkat lunak pendukung yaitu Perl untuk bahasa program yang digunakan dan sistem basis data MySQL yang digunakan untuk menyimpan seluruh data yang digunakan.

Pada *server mail* juga dipasang perangkat lunak Dovecot yang berfungsi sebagai *server* IMAP dan POP. Dovecot berfungsi sebagai *server* yang melayani akses pembacaan *email* melalui aplikasi *email* klien. Pengguna dapat mengakses *email* menggunakan layanan protokol IMAP dan POP. Dovecot akan menjalankan fungsi untuk autentikasi ketika pengguna akan melakukan akses pembacaan terhadap *email* yang dikirimkan kepadanya.

Router menggunakan sistem operasi Mikrotik RouterOS. Fitur Mikrotik yang digunakan dalam kaitan dengan penelitian adalah IP-Route. Dalam hal ini *Router* difungsikan sebagai pengatur rute paket data antara jaringan pada *server* dan jaringan pada komputer klien.

Komputer klien menggunakan sistem operasi Linux untuk keperluan operasi *desktop*. Perangkat lunak yang dipasang pada komputer klien yang berkaitan dengan penelitian ini adalah *email* klien. *Email* klien yang dipasang pada komputer klien terutama adalah Mozilla Thunderbird. Sebagai pembanding bisa juga pada komputer klien dipasang perangkat lunak *email* klien lain misalnya Rainloop *webmail*.

Komputer klien difungsikan juga sebagai komputer pengguna *email* ilegal, sehingga perlu dipasang perangkat lunak untuk menguji keamanan *server mail*. Pada penelitian digunakan perangkat lunak PHP untuk menyusun sekrip ilegal

login. Sekrip difungsikan untuk menguji keamanan *server mail*. Sekrip dapat digunakan untuk melakukan ilegal *login* menggunakan kamus *password* atau pun menggunakan metode *brute force password*.

4.3.3 Rancangan Tahapan Penelitian

Penelitian dilakukan dalam beberapa tahapan. Masing-masing tahapan yang dilakukan secara detail akan dibahas pada bagian selanjutnya. Tahapan-tahapan yang dilakukan pada saat melakukan penelitian dapat diuraikan secara ringkas sebagai berikut.

1. Menyiapkan jaringan
2. Menyiapkan *server mail*
3. Menyiapkan komputer klien.
4. Konfigurasi Mikrotik RouterOS pada *router*
5. Instalasi Centos 7 pada *server mail*
6. Instalasi Postfix pada *server mail*
7. Instalasi Dovecot pada *server mail*
8. Instalasi Fail2ban pada *server mail*
9. Instalasi Policyd pada *server mail*
10. Instalasi klien *mail* Thunderbird
11. Membuat program untuk melakukan ilegal login pada klien *mail*
12. Konfigurasi Postfix pada *server mail*

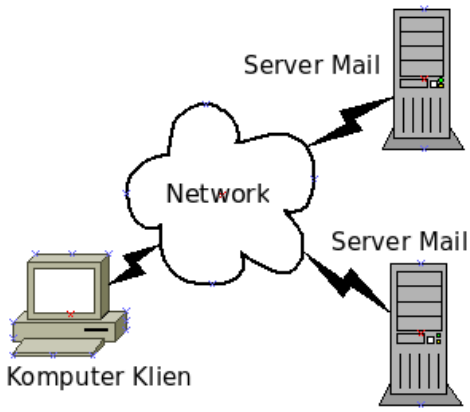
13. Konfigurasi Dovecot pada *server mail*
14. Konfigurasi Fail2ban pada *server mail*
15. Konfigurasi Policyd pada *server mail*
16. Menguji hasil konfigurasi menggunakan klien *mail* Thunderbird
17. Menguji gangguan keamanan *email* menggunakan sekrip ilegal login
18. Menguji cara mengatasi kegagalan Fail2ban menggunakan Policyd
19. Analisis terhadap hasil percobaan gangguan keamanan *email*
20. Membuat kesimpulan
21. Membuat laporan penelitian.

BAB 5 IMPLEMENTASI DAN PEMBAHASAN

5.1 Implementasi

5.1.1 Menyiapkan Jaringan

Konfigurasi jaringan yang diperlukan dalam penelitian ini melibatkan dua *server Mail*, satu klien, dan sistem pengaturan kabel jaringan komputer. Pada penelitian ini, sistem pengaturan kabel diabaikan karena komputer *server* dijalankan pada VirtualBox. *Server mail* dijalankan dalam bentuk mesin *virtual*, sedangkan klien dijalankan dari komputer *real* (komputer *host*).



Gambar 5.1 Topologi Logika Jaringan Untuk Penelitian

Penyiapan sistem jaringan yang digunakan pada penelitian sedikit berbeda

dengan apa yang dirancang pada bagian metodologi. Topologi yang digunakan pada perancangan adalah untuk topologi fisik menggunakan sistem jaringan fisik. Pada penelitian digunakan sistem jaringan pada VirtualBox sedemikian, sehingga sistem kabel dibentuk secara virtual. Hal demikian disebabkan sistem kabel juga disusun secara virtual. Untuk menyederhanakan topologi, peran fungsi *router* dilakukan bersama dalam mesin yang sama dengan komputer klien. Hal demikian tidak mempengaruhi esensi penelitian yang mana titik berat penelitian pada pengendalian *server mail* bukan pada bentuk topologi jaringan yang digunakan untuk menghubungkan antar peralatan jaringan.

5.1.2 Menyiapkan Server Mail

Server yang disusun pada penelitian dijalankan pada mesin virtual pada perangkat lunak VirtualBox. *Server mail* menggunakan sistem operasi Centos 7. Perangkat utama *server Mail* terdiri dari Postfix, Dovecot dan Fail2ban. Postfix berfungsi sebagai penyedia layanan SMTP sedangkan Dovecot berfungsi sebagai penyedia layanan IMAP dan POP3. Fail2ban digunakan untuk melakukan pengendalian akses pengguna pada *server mail*. *Server mail* juga memerlukan perangkat lunak MySQL untuk menyimpan pengguna *email* pada *server mail* pertama. Pada *server mail* kedua pengguna *email* dijadikan satu dengan pengguna sistem operasi.

5.1.2.1 Konfigurasi Postfix

Konfigurasi Postfix terutama terutama berkaitan dengan pengaturan TLS (*transport layer security*). TLS menyediakan komunikasi aman di Internet untuk hal-hal seperti *email*, faks Internet, dan transfer data lainnya. Ada sedikit perbedaan antara SSL 3.0 dan TLS 1.0, tetapi protokol yang digunakan pada dasarnya tetap sama. Adalah ide yang baik untuk diingat bahwa TLS berada pada Layer Aplikasi model OSI. Ini akan menghemat banyak frustrasi saat melakukan debug dan memecahkan masalah enkripsi terkait dengan TLS (Paul Szymanski, 2007).

Konfigurasi TLS memerlukan sertifikat SSL (*secure socket layer*) yang biasanya ada organisasi yang menyediakan sertifikat tersebut. Namun dalam penelitian ini, sertifikat SSL dibuat sendiri, sehingga tidak ada yang menjadi lembaga penjamin validitas.

Sertifikat SSL adalah *file* data kecil yang secara digital mengikat kunci kriptografi ke detail organisasi. Ketika instalasi pada *server web*, itu mengaktifkan gembok dan protokol HTTPS dan memungkinkan koneksi aman dari *server web* ke browser. Biasanya, SSL digunakan untuk mengamankan transaksi kartu kredit, transfer data dan login, dan baru-baru ini menjadi norma ketika mengamankan *browsing* situs media sosial (GlobalSign, 2018).

Konfigurasi utama Postfix disimpan pada *file* /etc/postfix/main.cf. Pengaturan berkaitan dengan fitur TLS adalah sebagai berikut

```
smtpd_use_tls = yes
```

```
smtpd_tls_key_file = /etc/postfix/postfix.key
smtpd_tls_cert_file = /etc/postfix/postfix.pem
```

Selanjutnya pengguna dipaksa untuk menggunakan *username* dan *password* ketika proses pengiriman *email*. Fitur ini perlu diaktifkan supaya *server mail* tidak digunakan oleh pengguna sembarang untuk mengirim *email* yang tidak bertanggung jawab.

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
```

Autentikasi pengguna *email* ditugaskan kepada *server* Dovecot, karena Postfix pada dasarnya tidak memiliki mekanisme autentikasi.

5.1.2.2 Konfigurasi Dovecot

Konfigurasi Dovecot terutama untuk dimaksudkan agar dapat memberi dukungan TLS pada Dovecot. Yang perlu diatur dalam untuk keperluan dukungan TLS adalah *file* /etc/pki/dovecot/dovecot-openssl.cnf.

```
C=ID
ST=DIY
L=Yogya
O=Dovecot
OU=Dovecot
CN=mail1.domain1.ku
emailAddress=postmaster@domain1.ku
nsCertType = server
```

Perlu diatur juga konfigurasi sertifikat SSL yang ada pada *file* /etc/dovecot/conf.d/10-ssl.conf terutama pada bagian berikut.

```
ssl = required
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
```

```
ssl_key = </etc/pki/dovecot/private/dovecot.key
```

Pengaturan ini untuk mengarahkan letak *file* sertifikat SSL yang dibuat untuk dukungan TLS.

5.1.2.3 Konfigurasi Fail2ban

Fail2ban digunakan untuk mengendalikan akses pengguna pada *server mail*. Pengendalian ini dimaksudkan untuk menambahkan aspek keamanan dari serangan pada *server mail*. Serangan yang umumnya sering dialami *server mail mail* adalah pembobolan password pengguna *email*. Apabila password berhasil dibobol, selanjutnya *server mail* digunakan untuk menyebarkan *email* spam ke jaringan Internet.

Spam *email*, juga dikenal sebagai *email* sampah, adalah pesan massal yang tidak diminta yang dikirim melalui *email*. Penggunaan spam telah semakin populer sejak awal sekitar tahu 1990 dan merupakan masalah yang dihadapi oleh sebagian besar pengguna *email*. Penerima spam sering memiliki alamat *email* mereka yang diperoleh oleh *spambot*, yang merupakan program otomatis yang merayapi Internet mencari alamat *email*. Spammer menggunakan spambot (robot spam) untuk membuat daftar distribusi *email*. Spammer biasanya mengirim *email* ke jutaan alamat *email*, dengan harapan hanya sejumlah kecil yang akan merespons atau berinteraksi dengan pesan tersebut (Margaret Rouse, 2017).

Konfigurasi Fail2ban terutama pada /etc/fail2ban/jail.conf. Namun dalam

pengaturannya tidak dianjurkan mengubah *file* konfigurasi tersebut. Konfigurasi dilakukan pada *file* yang diikutsertakan pada konfigurasi yaitu /etc/fail2ban/jail.local. Pengaturan dilakukan untuk mengendalikan akses pada layanan Postfix dan Dovecot. Berikut adalah pengaturan yang dilakukan.

```
[postfix-sasl]
enabled  = true
port     = smtp,465,submission,imap3,imaps,pop3,pop3s
logpath  = %(postfix_log)s
backend  = %(postfix_backend)s
bantime  = 120
findtime  = 120
maxretry = 3

[dovecot]
enabled = true
port    = pop3,pop3s,imap,imaps,submission,465,sieve
logpath = %(dovecot_log)s
backend = %(dovecot_backend)s
bantime  = 120
findtime  = 120
maxretry = 4
```

Konfigurasi hanya mengatur filter untuk *port*, letak *file* log, sekrip filter, waktu ban, selang waktu terjadi kegagalan *login* dan jumlah maksimum kegagalan yang diizinkan. Untuk Postfix yang diatur adalah filter [postfix-sasl], sedangkan untuk Dovecot digunakan filter [dovecot]. Pengaturan *port*, *logpath* dan *backend* menggunakan pengaturan yang telah disediakan oleh Fail2ban. Pengaturan bandtime dan findtime dibuat 120 detik. Artinya jeda waktu antar kegagalan yang tercatat adalah 120 detik. Waktu *ban* ditetapkan selama 120 detik. Jumlah kegagalan maksimum yang diizinkan untuk akses Postfix sebanyak 3 kali sedangkan untuk Dovecot sebanyak 4 kali. Pengaturan yang dilakukan ini, hanya untuk keperluan percobaan.

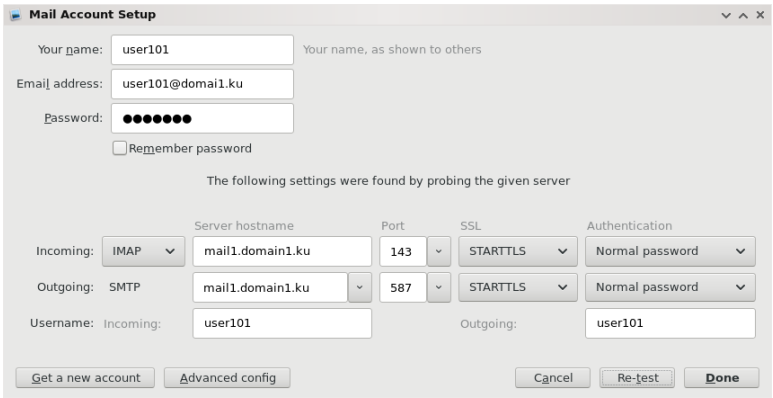
Dalam pelaksanaan percobaan, antara filter akses Postfix dan Dovecot

digunakan secara bergantian supaya pengamatan terhadap log akses lebih mudah dilakukan. Hal demikian perlu dilakukan mengingat isi *file log* yang bisa membingungkan apabila dua akses dilakukan filter bersama.

5.1.3 Menyiapkan Komputer Klien.

Komputer klien menggunakan sistem operasi Linux Mageia 6 yang dilengkapi perangkat lunak untuk akses *email* melalui Thunderbird dan Browser Mozilla Firefox. Thunderbird digunakan untuk menguji *server mail* dalam kaitan dengan pengguna legal dan ilegal. Browser digunakan untuk menjalankan sekrip PHP .Komputer klien juga dilengkapi dengan bahasa program PHP yang dilengkapi PHPMailer yang digunakan untuk membuat sekrip uji *server mail*. Dengan demikian komputer klien juga perlu dipasang *server web* Nginx.

Untuk menguji apakah konfigurasi TLS pada Postfix dan Dovecot dapat berfungsi dengan baik, digunakan Thunderbird sebagai berikut



Gambar 5.2 Hasil Uji Coba Fitur TLS Pada Postfix dan Dovecot

Selain menggunakan Thunderbird, *server mail* juga diuji dengan bantuan

sekrip PHP. Sekrip dirancang untuk melakukan penyebaran *spam* secara terprogram. Sekrip dibuat dengan bantuan PHPMailer yang punya beberapa fitur yang diperlukan untuk *login* pada *server mail* dan mengirim *email* secara terprogram.

Sekrip PHP yang digunakan untuk menguji *server mail* dalam mengatasi gangguan keamanan *password* adalah sebagai berikut.

```
<?php
require_once "PHPMailer/PHPMailerAutoload.php";
$mail = new PHPMailer;
$mail->isSMTP();
$mail->Host = "192.168.56.151";
$mail->SMTPAuth = true;
$mail->Username = "user101@domain1.ku";
$mail->Password = "user1011";
$mail->SMTPSecure = "tls";
$mail->Port = 587;
$mail->From = "user101@domain1.ku";
$mail->FromName = "user101";
$mail->addAddress("user201@domain2.ku", "user201");
$mail->Subject = "Subject pakai TLS";
$mail->Body = "Mail body";
//supaya tidak minta serifikat SSL
$mail->SMTPOptions = array(
  'ssl' => array(
    'verify_peer' => false,
    'verify_peer_name' => false,
    'allow_self_signed' => true
  )
);

if(!$mail->send())
    echo "Error ";
else
    echo "success";
?>
```

Pada sekrip tersebut diatur beberapa bagian yaitu nama *server mail*, username, password, alamat tujuan dan alamat asal, port, fitur TLS dan *email* yang dikirimkan. Selain itu juga perlu diatur bagian yang berkaitan dengan sertifikat SSL. Karena pada penelitian belum menggunakan sertifikat SSL resmi, perlu diatur supaya eksepsi permintaan sertifikat remi diabaikan. Kalau tidak

diatur demikian, maka pengiriman *email* menggunakan sekrip tersebut akan selalu mengalami kegagalan.

Sekrip akan memberikan pesan success apabila *email* berhasil dikirimkan, sebaliknya akan memberikan pesan error ketika *email* gagal dikirimkan. Pesan ini sebagai tanda untuk memudahkan berkaitan dengan pengamatan terhadap *log* yang dihasilkan oleh Postfix, Dovecot dan Fail2ban.

5.2 Pembahasan

Penelitian yang dilakukan adalah dengan pengamatan *file log* yang dihasilkan baik oleh Postfix, Dovecot maupun Fail2ban. *File log* yang dihasilkan Postfix dan Dovecot menjadi satu dalam *file* /var/log/maillog, sedangkan *log* yang dihasilkan oleh Fail2ban disimpan dalam *file* /var/log/fail2ban.log.

Apabila Thunderbird berhasil melakukan *login* pada *server mail*, maka pada *file maillog* akan muncul pesan-pesan berikut

```
Oct  2 08:11:31 mail1 dovecot: imap-login: Login:
user=<user101@domain1.ku>, method=PLAIN, rip=192.168.56.1,
lip=192.168.56.151, mpid=2422, TLS, session=<KvgRmjR3GgDAqDgB>
Oct  2 08:11:47 mail1 dovecot: imap-login: Login:
user=<user101@domain1.ku>, method=PLAIN, rip=192.168.56.1,
lip=192.168.56.151, mpid=2425, TLS, session=<vVUdmzR3JgDAqDgB>
Oct  2 08:11:48 mail1 dovecot: imap-login: Login:
user=<user101@domain1.ku>, method=PLAIN, rip=192.168.56.1,
lip=192.168.56.151, mpid=2428, TLS, session=<T5MfmzR3KADAqDgB>
Oct  2 08:11:49 mail1 dovecot: imap-login: Login:
user=<user101@domain1.ku>, method=PLAIN, rip=192.168.56.1,
lip=192.168.56.151, mpid=2431, TLS, session=<IrgzmzR3KgDAqDgB>
Oct  2 08:11:50 mail1 dovecot: imap-login: Login:
user=<user101@domain1.ku>, method=PLAIN, rip=192.168.56.1,
lip=192.168.56.151, mpid=2434, TLS, session=<A3xHmzR3LADAqDgB>
```

Pada *log* tercatat waktu *login* baik tanggal maupun jam, protokol yang

digunakan untuk *login*, pengguna yang *login* serta alamat IP asal pengguna melakukan *login*. Catatan ini sangat penting ketika ingin dilakukan penelusuran terhadap suatu *email* baik mengenai pengirim, dari mana *email* berasal dan kapan waktu pengiriman *email*.

Ketika *logout* dari *server mail*, pada *file maillog* juga akan tercatat parameter-parameter seperti pada saat *login*.

```
Oct  2 08:14:00 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Logged out in=223 out=699
Oct  2 08:14:00 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Logged out in=417 out=34382
Oct  2 08:14:00 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Logged out in=127 out=858
Oct  2 08:14:00 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Logged out in=296 out=1406
Oct  2 08:14:00 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Logged out in=122 out=890
```

Catatan ini berguna saat akan melakukan pelacakan siapa saja yang menggunakan *server mail* serta dari mana seorang pengguna menggunakan *server mail*. Pelacakan perlu dilakukan apabila terjadi gangguan pada *server mail*. Untuk pelacakan pengguna jaringan lokal, catatan ini berguna untuk menelusuri dari jaringan mana atau dari komputer mana gangguan terhadap *email* terjadi. Untuk pelacakan jaringan publik, catatan ini paling tidak dapat digunakan untuk mengetahui alamat IP yang menjadi sumber gangguan. Informasi ini dapat digunakan sebagai dasar untuk melakukan aksi terhadap asal gangguan *email*.

Apabila pengguna gagal *login* pada *server mail*, maka pada catatan *log* akan muncul sebagai berikut.

```
Oct  2 08:19:52 mail1 dovecot: imap-login: Disconnected (auth failed, 6
attempts in 46 secs): user=<user101@domain1.ku>, method=PLAIN,
rip=192.168.56.1, lip=192.168.56.151, TLS, session=<BPw5tTR3NADAqDgB>
Oct  2 08:21:13 mail1 dovecot: imap-login: Disconnected (auth failed, 2
```

```
attempts in 31 secs): user=<user101@domain1.ku>, method=PLAIN,
rip=192.168.56.1, lip=192.168.56.151, TLS, session=<X6jyujR3NgDAqDgB>
```

Kegagalan *login* ini adalah kegagalan untuk masuk *server* Dovecot. Kata kunci yang nantinya dipakai untuk melakukan filter kegagalan *login* adalah "auth failed". Kata kunci ini bisa dipakai oleh Fail2ban untuk melakukan pencatatan alamat IP yang akan dilakukan filter. Kegagalan ini bisa disebabkan tidak sengaja, misalnya pengguna salah memasukkan *password* atau juga karena pengguna lupa *password* kemudian mencoba beberapa *password*. Kegagalan bisa terjadi juga karena kesengajaan, misalnya pengguna sengaja ingin menembus keamanan *password* dengan cara mencoba banyak *password*.

Ketika pengguna akan melakukan pengiriman *email*, biasanya digunakan aplikasi *email* klien seperti Thunderbird. Langkah awal, Thunderbird akan *login* Dovecot. Ketika pengguna berhasil *login* Dovecot, maka pengguna dapat melihat isi *mailbox* yaitu daftar *email* masuk maupun keluar beserta isinya. Ketika pengguna berhasil *login* Dovecot, belum tentu pengguna dapat melakukan pengiriman *email* apabila *password* yang dimasukkan pada saat mengirim salah. Hal demikian bisa terjadi apabila pada proses masuk *server* dan mengirimkan *email* diatur sedemikian, sehingga pengguna harus selalu memasukkan *password*. Prosedur yang demikian dapat diatur dari aplikasi Thunderbird untuk mencegah penggunaan Thunderbird untuk menyebarkan *spam* apabila sistem komputer terinfeksi virus *email*.

Apabila pengguna berhasil *login* Dovecot dan berhasil dalam mengirim

email, maka pada *file maillog* akan muncul serangkaian catatan langkah-langkah pengiriman *email* sebagai berikut.

```
Oct  2 09:08:26 mail1 postfix/submission/smtpd[2688]: warning: hostname
gw.domain1.ku.56.168.192.in-addr.arpa does not resolve to address
192.168.56.1: Name or service not known
Oct  2 09:08:26 mail1 postfix/submission/smtpd[2688]: connect from
unknown[192.168.56.1]
Oct  2 09:08:33 mail1 postfix/submission/smtpd[2688]: 8719B2F45:
client=unknown[192.168.56.1], sasl_method=PLAIN,
sasl_username=user101@domain1.ku
Oct  2 09:08:33 mail1 postfix/cleanup[2690]: 8719B2F45: message-
id=<29358b28-f781-5bf6-32b7-4526ab9610ec@domain1.ku>
Oct  2 09:08:33 mail1 postfix/qmgr[2138]: 8719B2F45:
from=<user101@domain1.ku>, size=606, nrcpt=1 (queue active)
Oct  2 09:08:33 mail1 postfix/submission/smtpd[2688]: disconnect from
unknown[192.168.56.1]
Oct  2 09:08:33 mail1 dovecot: imap(user101@domain1.ku): Disconnected:
Disconnected in IDLE in=757 out=1851
Oct  2 09:08:38 mail1 postfix/smtp[2681]: 8719B2F45:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=5.1,
delays=0.06/0/5/0.06, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
6DF1F1EA6)
Oct  2 09:08:38 mail1 postfix/qmgr[2138]: 8719B2F45: removed
```

Log tersebut dihasilkan oleh Postfix ketika melakukan langkah-langkah pengiriman *email*. Postfix akan selalu mencatat semua aktivitas pengiriman *email* mulai saat koneksi *server mail* sampai *email* berhasil dikirimkan oleh *server mail*.

Apabila pengguna berhasil *login* Dovecot namun gagal dalam mengirim *email*, maka pada *file maillog* akan muncul serangkaian catatan langkah pengiriman *email* sebagai berikut.

```
Oct  2 09:12:27 mail1 postfix/submission/smtpd[2709]: connect from
unknown[192.168.56.1]
Oct  2 09:12:37 mail1 postfix/submission/smtpd[2709]: warning:
unknown[192.168.56.1]: SASL PLAIN authentication failed:
Oct  2 09:12:57 mail1 postfix/submission/smtpd[2709]: warning:
unknown[192.168.56.1]: SASL PLAIN authentication failed:
Oct  2 09:13:05 mail1 postfix/submission/smtpd[2709]: warning:
unknown[192.168.56.1]: SASL PLAIN authentication failed:
Oct  2 09:13:13 mail1 postfix/submission/smtpd[2709]: disconnect from
unknown[192.168.56.1]
```

Postfix akan mencatat aktivitas pengiriman *email* mulai saat *login* sampai dipastikan pengiriman *email* gagal. Kata kunci kegagalan pengiriman *email*

adalah "SASL PLAIN authentication failed" yang menunjukkan pengguna gagal autentikasi pengiriman *email*. Tanda ini bisa dipakai oleh Fail2ban untuk melalukan filter terhadap alamat IP pengirim *email*.

Kegagalan *login* pada Postfix juga dapat disebabkan karena tidak sengaja maupun karena sengaja. Kegagalan *login* yang tidak sengaja bisa disebabkan salah menekan papan ketik atau pengguna lupa *password* dan mencoba banyak *password*. Kegagalan yang sengaja terjadi karena pengguna ingin menembus keamanan *password* untuk tujuan yang tidak baik, misalnya untuk penyebaran *email spam*.

Kegagalan *login* dalam penggunaan *server mail* harus dikendalikan terutama kegagalan yang sengaja untuk menembus keamanan *password*. Salah satu usaha yang sering digunakan untuk menembus keamanan *password* adalah melakukan *brute force*. Cara ini dilakukan dengan mencoba banyak *password* menggunakan aplikasi tertentu. *Password* yang dicoba bisa berasal dari kamus *password* atau *password* acak. Pengiriman in dilakukan dengan cara yang sangat cepat menggunakan teknik program tertentu.

Pada penelitian ini dicoba untuk mengendalikan gangguan keamanan *password* pada *email* dengan cara mengendalikan jumlah kegagalan *login*. Supaya dapat mengawasi jumlah kegagalan *login* yang dicatat pada catatan *log server* digunakan teknologi Fail2ban. Fail2ban dapat digunakan untuk memonitor, mencatat dan membuat aksi untuk merespons kegagalan yang terjadi pada *server mail* dengan cara mengawasi *file log.*

5.2.1 Pengujian Fail2ban Untuk Pengendalian Akses Dovecot

Selanjutnya dibahas penerapan Fail2ban untuk melakukan pengendalian akses Dovecot yaitu pada saat pengguna *login* aplikasi Thunderbird. Bagian yang dikenakan konfigurasi pada Fil2ban adalah bagian [dovecot]. Konfigurasi Fail2ban diatur dengan cara membatasi jumlah kegagalan *login* sebanyak 4 kali, batas rentang waktu minimal boleh terjadi kegagalan *login* sebesar 120 detik dan waktu *ban* dilakukan selama 120 detik. Setelah 120 detik, dilepaskan filter *ban* terhadap suatu alamat IP. Ban akan dilakukan pada satu alamat IP yang tidak termasuk alamat IP pengecualian.

Pada saat layanan Fail2ban dijalankan akan terbentuk aturan Firewall Iptables awal yang berkaitan dengan layanan Dovecot. Firewall akan membentuk target f2b-dovecot yang digunakan untuk melakukan filter pada *port* yang menjadi layanan Dovecot yaitu protokol yang berkaitan dengan pembacaan *email* seperti IMAP, IMAPS, POP3 maupun POP3S.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-dovecot  tcp  --  0.0.0.0/0          0.0.0.0/0            multiport
dports 110,995,143,993,587,465,4190

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-dovecot (1 references)
target     prot opt source               destination
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Ketika pengguna melakukan kegagalan *login* sebanyak 4 kali dalam rentang waktu 120 detik, maka Fail2ban akan melakukan proses *ban* terhadap alamat IP pengguna. Apabila alamat IP pengguna dikenakan ban, maka pengguna tidak bisa melakukan koneksi ke *server mail*. Log proses *ban* oleh Fail2ban dapat dilihat pada *file* fail2ban.log.

```
2018-10-02 09:27:07,169 fail2ban.filter         [2901]: INFO    [dovecot]
Found 192.168.56.1
2018-10-02 09:27:43,157 fail2ban.filter         [2901]: INFO    [dovecot]
Found 192.168.56.1
2018-10-02 09:27:57,520 fail2ban.filter         [2901]: INFO    [dovecot]
Found 192.168.56.1
2018-10-02 09:28:53,328 fail2ban.filter         [2901]: INFO    [dovecot]
Found 192.168.56.1
2018-10-02 09:28:53,625 fail2ban.actions        [2901]: NOTICE  [dovecot]
Ban 192.168.56.1
2018-10-02 09:30:54,099 fail2ban.actions        [2901]: NOTICE  [dovecot]
Unban 192.168.56.1
```

Dapat dilihat pada catatan log, Fail2ban berhasil mendeteksi kegagalan *login* Dovecot sebanyak 4 kali dari jam 09:27:07 sampai jam 09:28:53. Segera setelah terjadi kegagalan yang keempat, pada jam 09:28:53 Fail2ban melakukan proses *ban* untuk alamat IP 192.168.56.1 yang merupakan alamat IP asal yang dipakai pengguna. Pada jam 09:30:54, Fail2ban melakukan proses *unban* untuk melepas blok alamat IP 192.168.56.1 yaitu rentang waktu 120 detik setelah proses *ban* ditetapkan.

Kejadian yang mana pengguna gagal melakukan *login* tercatat 4 kali yang memicu Fail2ban untuk melakukan *ban* terhadap alamat IP. Proses *ban* dilakukan dengan cara menambah aturan Firewall untuk melakukan penolakan (REJECT) koneksi dai alamat IP yang terkena aturan ban. Tindakan aksi *ban* dilakukan oleh Fail2ban dengan cara membuat aturan Firewall baru untuk melakukan blok

sebagai berikut.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-dovecot  tcp  --  0.0.0.0/0          0.0.0.0/0            multiport
dports 110,995,143,993,587,465,4190

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-dovecot (1 references)
target     prot opt source               destination
REJECT     all  --  192.168.56.1         0.0.0.0/0            reject-
with icmp-port-unreachable
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Pada aturan Firewall yang baru terlihat bahwa alamat IP 192.168.56.1 terkena *ban* yang menyebabkan semua koneksi dari alamat IP tersebut ditolak oleh *server mail*. Penolakan koneksi oleh *server* tentunya akan menghentikan aktivitas *login* yang dilakukan oleh aplikasi klien.

Proses *ban* terhadap alamat IP klien diterapkan selama 120 detik sesuai aturan yang ditetapkan pada bantime yang terdapat pada *file* konfigurasi jail.local. Setelah waktu 120 detik terlampaui, Fail2ban akan melepaskan *ban* pada alamat IP tersebut melalui proses unban. Pada saat proses *unban* terjadi, aturan Firewall yang melakukan penolakan alamat IP tersebut akan dihapus.

5.2.2 Pengujian Fail2ban Untuk Pengendalian Akses Posfix

Supaya dapat melakukan pengendalian akses Postfix, bagian [postfix-sasl] pada *file* jail.local harus diaktifkan. Fail2ban diatur untuk membatasi jumlah kegagalan *login* sebanyak 3 kali, batas rentang waktu minimal boleh terjadi

kegagalan *login* sebesar 120 detik dan waktu *ban* dilakukan selama 120 detik. Ban akan dilakukan pada satu alamat IP yang tidak termasuk alamat IP pengecualian.

Selanjutnya dilakukan restart pada layanan Fail2ban. Fail2ban akan membentuk aturan Firewall awal untuk pengendalian akses Postfix. Iptables akan membentuk target f2b-postfix yang digunakan untuk melakukan filter layanan Postfix yang meliputi SMPT, SMTPS maupun *submission email*. Aturan Firewall yang terbentuk adalah sebagai berikut.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-postfix  tcp  --  0.0.0.0/0          0.0.0.0/0            multiport
dports 25,465,587,220,993,110,995

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-postfix (1 references)
target     prot opt source               destination
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Ketika pengguna melakukan kesalahan saat melakukan proses pengiriman *email*, maka akan dicatat oleh Fail2ban. Apabila kesalahan dilakukan sebanyak 3 kali, maka Fail2ban melakukan proses *ban* terhadap alamat IP asal pengguna. Pada catatan *log* Fail2ban akan muncul pesan-pesan kesalahan sebagai berikut.

```
2018-10-02 10:04:53,245 fail2ban.filter         [3283]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 10:05:05,328 fail2ban.filter         [3283]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 10:05:19,086 fail2ban.filter         [3283]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 10:05:19,911 fail2ban.actions        [3283]: NOTICE  [postfix-
sasl] Ban 192.168.56.1
2018-10-02 10:07:20,341 fail2ban.actions        [3283]: NOTICE  [postfix-
sasl] Unban 192.168.56.1
```

Pada *log* terlihat bahwa Fail2ban mencatat telah terjadinya kegagalan

login sebanyak 3 kali dari jam 10:04:53 sampai jam 10:05:19 yang masuk dalam rentang 120 detik. Pada jam 10:05:19 dilakukan *ban* pada alamat IP 192.168.56.151 yaitu segera setelah pengguna melakukan kesalahan yang ketiga. Pada saat Fail2ban melakukan *ban* pada alamat IP tersebut, pada aturan Firewall muncul filter untuk *ban* alamat IP tersebut.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-postfix-sasl  tcp  --  0.0.0.0/0            0.0.0.0/0
multiport dports 25,465,587,220,993,110,995

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-postfix-sasl (1 references)
target     prot opt source               destination
REJECT     all  --  192.168.56.1         0.0.0.0/0            reject-
with icmp-port-unreachable
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Pada saat terjadi ban, pada aturan Firewall muncul aturan yang menyebabkan koneksi dari alamat IP 192.168.56.1 ke *server mail* ditolak (REJECT). Dengan aturan ini, usaha pengiriman *email* melalui *server mail* dari alamat IP tersebut akan mengalami kegagalan. Kegagalan ini terjadi pada saat autentikasi *password* untuk pengiriman *email*.

Apabila keadaan *ban* pada suatu alamat IP dibebaskan (unban), Fail2ban akan menghapus aturan filter penolakan koneksi dari alamat IP 192.168.56.1. Penghapusan aturan ini tentunya menyebabkan pengguna dapat kembali lagi melakukan usaha pengiriman *email*.

Selang waktu antara terjadi *ban* dan *unban* oleh Fail2ban bisa diatur sesuai situasi dan kondisi. Apabila *server mail* sering mengalami gangguan keamanan

password, selang waktu *ban* dan *unban* bisa dibuat lebih besar. Namun apabila *server mail* jarang mengalami gangguan, selang waktu bisa diatur lebih kecil. Perlu dicari waktu *ban* yang paling sesuai dengan situasi dan kondisi *server mail*.

5.2.3 Pengujian Server Mail Menggunakan Sekrip PHPMailer

Pengujian *server mail* menggunakan PHPMailer meniru yang dilakukan *spammer* dalam mengirim *email spam*. *Email spam* tidak mungkin dikirimkan secara manual satu demi satu menggunakan aplikasi seperti Thunderbird. *Email spam* dikirimkan menggunakan sekrip program. Walaupun tidak persis sama, namun penggunaan PHPMailer dapat mewakili apa yang dilakukan oleh *spammer*.

Berikut ini adalah catatan *file maillog* berkaitan dengan uji yang dilakukan menggunakan PHPMailer.

```
Oct  2 10:58:31 mail1 postfix/submission/smtpd[3623]: warning: hostname
gw.domain1.ku.56.168.192.in-addr.arpa does not resolve to address
192.168.56.1: Name or service not known
Oct  2 10:58:31 mail1 postfix/submission/smtpd[3623]: connect from
unknown[192.168.56.1]
Oct  2 10:58:33 mail1 postfix/submission/smtpd[3623]: warning:
unknown[192.168.56.1]: SASL PLAIN authentication failed:
Oct  2 10:58:33 mail1 postfix/submission/smtpd[3623]: disconnect from
unknown[192.168.56.1]
```

Kode kesalahan yang ditampilkan pada catatan *log* tidak membedakan antara kesalahan *password* yang dihasilkan oleh PHPMailer dan Thunderbird. Dengan demikian Postfix tidak memperhatikan jenis aplikasi *email* klien yang digunakan oleh pengguna.

Berikut adalah catatan fail2ban.log ketika pengguna melakukan kesalahan akses sebanyak tiga kali yang menyebabkan Fail2ban menjalankan aksi untuk melakukan *ban* pada alamat IP pengguna.

```
2018-10-02 11:04:23,349 fail2ban.filter         [3801]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 11:04:36,198 fail2ban.filter         [3801]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 11:04:43,177 fail2ban.filter         [3801]: INFO    [postfix-
sasl] Found 192.168.56.1
2018-10-02 11:04:43,562 fail2ban.actions        [3801]: NOTICE  [postfix-
sasl] Ban 192.168.56.1
2018-10-02 11:06:43,929 fail2ban.actions        [3801]: NOTICE  [postfix-
sasl] Unban 192.168.56.1
```

Metode coba-coba secara manual untuk menembus keamanan *password* menggunakan aplikasi klien *mail*, relatif bisa diatasi menggunakan teknologi Fail2ban sederhana. Apabila *spammer* tidak mempunyai cadangan alamat IP yang banyak, maka teknologi Fail2ban relatif dapat digunakan untuk mengendalikan gangguan terhadap *server mail*.

Metode *brute force* yang digunakan untuk menembus keamanan *password* bisa berhasil apabila *spammer* menggunakan mesin robot untuk menembus keamanan *password*. Selain itu *spammer* punya banyak cadangan alamat IP sebagai sumber untuk menembus keamanan *password*. *Spammer* menggunakan alamat IP secara acak bergantian sedemikian, sehingga pada akhirnya keamanan *password* dapat ditembus.

Selain *spammer*, gangguan *server mail* dapat muncul dari pengguna aplikasi klien *mail* yang terjangkiti virus *email*. Aplikasi klien *mail* punya fitur untuk menyimpan *username* dan *password* pengguna. Fitur ini memudahkan pengguna untuk masuk *server mail* dan mengirim *email*. Bahkan dengan adanya

fitur ini, notifikasi *email* dapat diaktifkan, sehingga pengguna merasa nyaman dalam menggunakan klien *mail*. Namun di samping kenyamanan yang diperoleh, tersimpan bahaya bagi *server mail* dengan adanya fitur simpan *password*. Biasanya, klien *mail* akan minta konfirmasi, apakah *password* pengguna akan disimpan atau tidak.

Virus *email* berusaha masuk ke *server mail* menggunakan *password* yang disimpan pengguna pada aplikasi. Virus *email* akan berlaku sebagai *spammer* ketika berhasil mendapatkan *username* dan *password* yang disimpan pada klien *mail*. Virus akan mengirim *email* secara acak menggunakan *username* dan *password* yang diperoleh secara ilegal. Hal demikian menyebabkan *server mail* akan dianggap penyebar *spam* oleh penerima *email*. Pada akhirnya bisa menyebabkan *server mail* dilakukan blok oleh *server mail* lain. Hal demikian tentunya sangat merugikan *server mail*.

Apabila berhasil mendapatkan *username* dan *password*, maka virus akan berlaku sebagai *spammer* untuk menyebarkan *email spam* secara acak menuju alamat *email* acak atau alamat *email* sesuai data yang dipunyai virus. Dalam kaitan ini, virus *email* berlaku seperti pengirim *email* legal pada *server mail*, karena menggunakan *username* dan *password* yang legal.

Untuk mengendalikan *server mail* ketika keamanan *password* berhasil ditembus, salah satunya dengan cara membatasi kecepatan pengiriman *email* menuju domain luar. Pengendalian pengiriman *email* dapat dilakukan menggunakan bantuan perangkat lunak tambahan yang dipasang untuk

melengkapi Postfix. Salah satu perangkat lunak yang dapat dipakai untuk mengatur kecepatan *email* keluar diantarnya adalah Policyd.

Policyd dapat mengatur jumlah pengiriman *email* tiap satuan waktu tertentu. Pengaturan Policyd dapat dengan mudah dilakukan melalui antarmuka *web*. Namun ternyata, berdasarkan penelitian awal, Policyd tidak dapat mendeteksi *email spam* yang dikirimkan oleh virus, sehingga apabila *server mail* terkena serangan, virus *email* masih dapat leluasa mengirimkan *email spam* secara acak keluar dari *server mail*. Hal demikian menyebabkan *server mail* dapat dikenakan blok oleh *server mail* tujuan.

Cara yang dicoba untuk mengatasi pengiriman acak *email spam* pada penelitian ini, dilakukan langsung pada Postfix. Postfix punya fasilitas untuk mengatur kecepatan pengiriman *email* keluar melalui pengaturan yang dituliskan pada *file* konfigurasi. Berikut adalah pengaturan pada beberapa *file* konfigurasi. Yang pertama adalah *file* /etc/postfix/transport yang mengatur pembentukan filter untuk mendeteksi tubuh *email*.

```
/user201\@domain2\.ku$/        smtp-domain2:
```

Sebagai bahan percobaan, diuji untuk melakukan filter pada *email* yang ditujukan pada alamat user201@domain2.ku. *Email* yang keluar menuju alamat tersebut dilakukan filter sesuai aturan transpor yang diberi tanda smtp-domain2. *Email* yang keluar menuju alamat tersebut akan dikenakan aturan antrean pengiriman *email*.

Selanjutnya pada *file* konfigurasi /etc/postfix/master.cf ditentukan

beberapa pengaturan pengiriman *email* menggunakan protokol SMTP pada tanda yang sudah diberikan dari *file* transpor.

```
smtp-domain2 unix - - n - 1 smtp
       -o syslog_name=smtp-domain2
```

Pengaturan utama pada *file* konfigurasi /etc/postfix/main.cf untuk pengendalian kecepatan pengiriman *email*. Beberapa pengarah perlu diatur untuk mendapatkan pengaturan kecepatan pengiriman yang diinginkan. Nama pengarah disesuaikan dengan nama transpor yang akan diatur. Dalam pengaturan ini diatur kecepatan transpor dengan nama smtp-domain2.

```
smtp-domain2_destination_rate_delay = 60s
smtp-domain2_destination_concurrency_limit = 1
smtp-domain2_destination_recipient_limit = 2
smtp-domain2_initial_destination_concurrency = 1
```

Aturan tersebut menentukan bahwa *email* yang mengikuti aturan transpor smtp-domain2 dikenakan kecepatan penundaan sebesar 60 detik untuk tiap satu *email* yang akan dikirimkan. Dengan ungkapan lain, Postfix akan mengirimkan sebanyak satu *email* tiap 60 detik. Dengan demikian kecepatan pengiriman *email* ke alamat yang diatur dengan transpor smtp-domain2 sebesar 1 *email* tiap 60 detik. Jika terdapat banyak *email* yang akan dikirimkan pada alamat tersebut, maka *email* akan mengalami antrean.

Berikut ini adalah hasil catatan *file maillog* pada percobaan pengiriman lima *email* dalam waktu yang relatif bersamaan, sebagai simulasi terjadi pengiriman *email spam* .

```
Oct  7 16:46:55 mail1 postfix/submission/smtpd[2855]: 10E1530C7:
client=unknown[192.168.56.1], sasl_method=PLAIN,
sasl_username=user101@domain1.ku
```

```
Oct  7 16:46:55 mail1 postfix/cleanup[2866]: 10E1530C7: message-
id=<500bb76fe438a97fbdbc980fb0975936@localhost.localdomain>
Oct  7 16:46:55 mail1 postfix/qmgr[2776]: 10E1530C7:
from=<user101@domain1.ku>, size=549, nrcpt=1 (queue active)
Oct  7 16:46:55 mail1 postfix/submission/smtpd[2855]: disconnect from
unknown[192.168.56.1]
Oct  7 16:46:56 mail1 postfix/submission/smtpd[2855]: warning: hostname
gw.domain1.ku.56.168.192.in-addr.arpa does not resolve to address
192.168.56.1: Name or service not known
Oct  7 16:46:56 mail1 postfix/submission/smtpd[2855]: connect from
unknown[192.168.56.1]
Oct  7 16:46:56 mail1 postfix/submission/smtpd[2855]: B8D6A30CE:
client=unknown[192.168.56.1], sasl_method=PLAIN,
sasl_username=user101@domain1.ku
Oct  7 16:46:56 mail1 postfix/cleanup[2866]: B8D6A30CE: message-
id=<03e081ab909c4c8af46d1c051fa1bfbb@localhost.localdomain>
Oct  7 16:46:56 mail1 postfix/qmgr[2776]: B8D6A30CE:
from=<user101@domain1.ku>, size=549, nrcpt=1 (queue active)
Oct  7 16:46:56 mail1 postfix/submission/smtpd[2855]: disconnect from
unknown[192.168.56.1]
Oct  7 16:47:16 mail1 smtp-domain2/smtp[2806]: C6AB92B3C:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=24,
delays=0.15/19/5/0.03, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
5B902245D)
Oct  7 16:47:16 mail1 postfix/qmgr[2776]: C6AB92B3C: removed
Oct  7 16:48:22 mail1 smtp-domain2/smtp[2806]: C17A530C5:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=88,
delays=0.06/83/5/0.03, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
750B719A0)
Oct  7 16:48:22 mail1 postfix/qmgr[2776]: C17A530C5: removed
Oct  7 16:49:27 mail1 smtp-domain2/smtp[2806]: 689DD304C:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=153,
delays=0.08/148/5/0.31, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
8E07D245D)
Oct  7 16:49:27 mail1 postfix/qmgr[2776]: 689DD304C: removed
Oct  7 16:50:32 mail1 smtp-domain2/smtp[2806]: 10E1530C7:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=217,
delays=0.07/212/5/0.03, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
EB0EB245D)
Oct  7 16:50:32 mail1 postfix/qmgr[2776]: 10E1530C7: removed
Oct  7 16:51:37 mail1 smtp-domain2/smtp[2806]: B8D6A30CE:
to=<user201@domain2.ku>, relay=domain2.ku[192.168.57.152]:25, delay=281,
delays=0.11/276/5/0.04, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as
0FF0F245D)
Oct  7 16:51:37 mail1 postfix/qmgr[2776]: B8D6A30CE: removed
```

Ketika pertama kali *server mail* menerima *email* yang akan dikirimkan menuju alamat yang dikenakan aturan transpor, maka *email* akan dimasukkan dalam antrean. *Email* yang datang pertama pada *server mail*, akan segera dikirimkan ke alamat tujuan. Penundaan yang terjadi pada *email* pertama adalah sebesar 24 detik. Penundaan pengiriman *email* disebabkan adanya beberapa langkah yang harus dilalui oleh suatu *email*, mulai saat dilakukan proses sampai

email tersebut dikirimkan.

Email yang kedua baru dikirimkan setelah mengalami penundaan selama 88 detik. Pengiriman baru dilakukan 64 detik setelah *email* pertama dilakukan. *Email* ketiga baru dikirimkan setelah ditunda 153 detik atau 65 detik setelah pengiriman *email* kedua. *Email* keempat dikirimkan setelah ditunda selama 217 detik atau 64 detik setelah *email* ketiga dikirimkan. *Email* kelima dikirimkan setelah ditunda selama 281 detik atau 64 detik setelah *email* keempat dikirimkan. Dengan demikian pengaturan kecepatan pengiriman *email* sebesar 1 *email* tiap 60 detik berhasil dilakukan.

Pengaturan kecepatan pengiriman *email* sangat mempengaruhi keputusan *server mail* lain yang menjadi tujuan untuk menentukan apakah *email* yang diterima merupakan *email spam* atau bukan. Jika *email* yang dikirimkan oleh *server mail* kepada *server mail* lain terlalu tinggi, maka *server mail* yang menjadi tujuan dapat menganggap sedang mendapatkan serangan *email* spam. *Server mail* tujuan dapat melakukan blok terhadap *server mail* sumber *email* spam.

Ada *server mail* yang sangat ketat dalam melakukan blok pada penyebar *email spam* (misalnya gmail.com) sedemikian, sehingga apabila suatu *server mail* dianggap sebagai penyebar *email spam*, maka dilakukan blok yang cukup lama. Namun *server mail* lain cukup longgar ketika melakukan blok pada suatu *server mail* sumber *spam* (misal yahoo.com) sedemikian, sehingga proses blok hanya dilakukan selama beberapa jam saja.

Dengan demikian perlu diatur penyesuaian kecepatan pengiriman *email*

yang menuju pada suatu tujuan. *Email* yang menuju gmail.com, perlu diatur agar kecepatan pengiriman *email* dibuat cukup rendah, namun tidak mengganggu kelancaran pengiriman *email*. Sebaliknya, *email* yang menuju alamat selain gmail.com dapat diatur agar kecepatan pengiriman *email* bisa diatur lebih tinggi, namun tetap dalam batas wajar. Hal demikian perlu pengaturan supaya *server mail* tidak dianggap sebagai penyebar *spam*, meskipun dalam keadaan sedang menerima serangan penyebar *email spam*.

BAB 6 KESIMPULAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil pembahasan dan percobaan dalam penelitian pengendalian *server mail* menggunakan teknologi Fail2ban adalah sebagai berikut.

1. Penelitian berhasil menyusun metode untuk melindungi *server mail* dari gangguan yang berkaitan dengan *email*.
2. Penelitian berhasil menerapkan perangkat lunak Fail2ban untuk melindungi *server mail* dari gangguan keamanan terutama gangguan *brute force password*.
3. Penelitian belum sampai menyusun metode deteksi untuk melihat apakah keamanan *server mail* berhasil diterobos. Namun demikian, penelitian berhasil menyusun metode untuk mengatasi pengguna ilegal yang berhasil menerobos keamanan *password* tanpa memperhatikan cara yang digunakan pengguna untuk menerobos keamanan *password* tersebut.
4. Penelitian berhasil menerapkan metode pengamanan *password* pada operasi *server mail* sesungguhnya yang mana diperoleh bahwa Policyd tidak berhasil mengatasi serangan oleh virus *email*. Untuk mengatasi

serangan virus *email*, sebaiknya digunakan pengaturan kecepatan pengiriman *email* melalui konfigurasi Postfix.

6.2 Saran

Saran yang diajukan untuk pengembangan dan penelitian lebih lanjut dari penelitian ini adalah sebagai berikut.

- Penelitian untuk menyusun metode deteksi untuk melihat apakah keamanan *server mail* berhasil diterobos
- Penelitian untuk mengatasi secara dini adanya serangan terhadap *server mail* berdasarkan *file log* yang dihasilkan Dovecot, Postfix maupun Fail2ban.

Daftar Pustaka

AllWorldIT, 2018, *policyd*, https://wiki.policyd.org/

Brent R. Matzelle, 2017, *PHPMailer - A full-featured email creation and transfer class for PHP*, https://github.com/PHPMailer/PHPMailer/wiki/Tutorial

Chris C, 2016, *fail2ban*, https://www.fail2ban.org

Chris C, 2018, *How to harden a server with fail2ban*, https://www.a2hosting.com/kb/security/hardening-a-server-with-fail2ban

Cyril Jaquier, 2017, *fail2ban - bans IP that makes too many password failures*, Linux Documentation

Dan Nanni, 2013, *How to protect SSH server from brute force attacks using fail2ban*, http://xmodulo.com/how-to-protect-ssh-server-from-brute-force-attacks-using-fail2ban.html

Dandy Pramana Hostiadi, 2016, *Implementasi Pengamanan E-mail Menggunakan Pretty Good Privacy Pada Zimbra Mail Server*, SISITI : Seminar Ilmiah Sistem Informasi dan Teknologi Informasi, http://epublications.dipanegara.ac.id/index.php/sisiti

Elle Krout, 2017, *Use Fail2ban to Secure Your Server*, https://www.linode.com/docs/security/using-fail2ban-for-security/

Galih Dwiyan Prakoso, dkk., 2017, *Implementasi Kemanan Mail Server Zimbra Menggunakan Spamassassin dan Whitlist Pada Linux Centos 7*, e-Proceeding of Applied Science, https://openlibrary.telkomuniversity.ac.id/home/epublication/id/132.html

GlobalSign, 2018, *What is an SSL Certificate?*, https://www.globalsign.com/en/ssl-information-center/what-is-an-ssl-certificate/

Kyle D , 2003, *Postfix: The Definitive Guide*, O'Reilly

Margaret Rouse, 2017, *email spam*, https://searchsecurity.techtarget.com/definition/spam

Matthew Setter, 2017, *How To Protect Against Brute Force Logins With Fail2Ban*, https://matthewsetter.com/bruteforce-protection-with-fail2ban/

Mika Larramo, 2018, *What is a Mail Server and How Does it Work?*, http://www.samlogic.net/articles/mail-server.htm

Nethesis Srl, 2017, *Fail2ban*, http://docs.nethserver.org/en/v7/fail2ban.html

Nikolai Lusan, 2017, *Policyd v2 (Clurbringer) Postfix policy server*, Linux Documentation

Paul Szymanski, 2007, *What is Transport Layer Security protocol?*, https://www.networkworld.com/article/2303073/lan-wan/lan-wan-what-is-transport-layer-security-protocol.html

Postfix, 2018, *Postfix feature overview*, http://www.postfix.org/features.html
Rusty Russell, 2017, *iptables - administration tool for IPv4 packet filtering and NAT*, Linux Documentation
Supriyo Biswas, 2018, *Protecting SSH with Fail2ban*, https://www.booleanworld.com/protecting-ssh-fail2ban/
Supriyo Biswas, 2018, *Protecting SSH with Fail2ban*, https://www.booleanworld.com/protecting-ssh-fail2ban/
Timo Sirainen, 2017, *dovecot - a secure and highly configurable IMAP and POP3 server*, Linux Documentation
Wietse Venema, 2017, *Postfix control program*, Linux Documentation
Zimbra, Inc, 2016, *How-to for CBPolicyd*, https://wiki.zimbra.com/wiki/Cluebringer_Policy_Daemon

LAMPIRAN

Curriculum Vitae

Nama	: W A G I T O, S.T., M.T.
Umur	: 46 tahun
Pangkat / Golongan	: Pembina Tk 1 / IV B
Jabatan Fungsional	: Lektor Kepala
Riwayat Pendidikan	
SD	: 1983
SMP	: 1986
SMA	: 1989
Sarjana Teknik	: 1994
Magister Teknik	: 1999
Alamat	: Suryoputran Pb III / 44 Yogyakarta 55131